Seminar Nasional Teknik Informatika (SANTIKA) 2013
Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Pembangunan Nasional "Veteran" Jawa Timur
18 September 2013. pp 51-56

PENGENALAN KARAKTER PADA PROSES DIGITALISASI DOKUMEN MENGGUNAKAN COSINE SIMILARITY

Wahyu S. J. Saputra, Faisal Muttaqin
Teknik Informatika, Fakultas Teknologi Industri, Universitas Pembangunan Nasinal "Veteran" Jatim
E-mail: Wahyu.s.j.saputra@gmail.com, f4154l_m@yahoo.co.id

Abstrak: Proses perubahan dokumen teks menjadi dokumen digital membutuhkan tahapan yang disebut sebagai pengenalan karakter. Pada umumnya proses pengenalan karakter membutuhkan proses *learning* dalam implementasinya, sehingga diperlukan waktu khusus. Proses similaritas telah digunakan dalam pengolahan citra. Pada penelitian ini diusulkan sebuah metode pengenalan karakter dengan menggunakan prinsip similaritas. Masukan dari metode ini adalah citra karakter yang selanjutnya dilakukan proses *resizing*, *skeletoning*, segmentasi, dan *similarity*. Metode similaritas yang digunakan adalah *Cosine Similarity* dengan membandingkan matriks dari citra masukan dengan citra yang terdapat dalam *database template*. Keluaran dari sistem adalah karakter yang terpilih dari nilai tertinggi yang didapatkan dari proses perbandingan similaritas dengan seluruh data dalam *database*.

Kata Kunci: Citra, Karakter, Tulisan, *skeletoning*, Similaritas, *Cosine Similarity*.

1. PENDAHULUAN

Pengenalan karakter merupakan tahapan yang paling penting dalam perubahan dokumen teks yang tercetak, menjadi dokumen digital. Pengenalan karakter juga diperlukan untuk hal yang lain selain digitalisasi dokumen, salah satunya adalah pengenalan NOPOL (Nomor Polisi) kendaraaan [1]. Pengenalan karakter merupakan tahap yang dilakukan setelah gambar digital dari teks didapatkan. Pengenalan karakter menjadi semakin kompleks ketika dokumen yang diproses merupakan sebuah dokumen yang terdiri dari banyak karakter. NOPOL kendaraan terdapat kurang lebih 15 sampai 20 karakter dengan dua ukuran yang berbeda, lain halnya dengan dokumen teks yang terdiri dari ratusan bahkan ribuan karakter.

Sebuah algoritma baru diusulkan dengan menggunakan algoritma optimasi untuk mengenali karakter dalam sebuah dokumen. Algoritma optimasi ditambahkan untuk meningkatkan akurasi dan presisi yang ditambahkan pada tahap preprocessing. Optimasi dilakukan dengan cara melakukan modifikasi pada gambar (*template*) dari dokumen. Modifikasi yang dilakukan antara lain rotasi dan translasi. Pengenalan karakter didasarkan pada contoh (*template*) karakter diperlukan optimasi pada contoh karakter untuk meningkatkan akurasi dan presisi [2].

Algoritma pengenalan karakter terus dikembangkan dan diaplikasikan, salah satu aplikasi dari pengembangan algoritma pengenalan karakter adalah dengan menggunakan algoritma pengenalan karakter tidak hanya untuk mengenali NOPOL kendaraan bermotor, namun juga mengenali sebuah dokumen sejarah. Dengan menggunakan basis data dari beragam jenis karakter dalam dokumen sejarah yang digunakan sebagai data pelatihan sistem, maka sistem mampu mengenali dokumen sejarah berdasarkan karakternya. Pada algoritma ini juga dilakukan proses pengelompokan (*clustering*) untuk karakter yang memiliki bentuk yang mirip. *Database* yang telah dibentuk digunakan untuk mengenali karakter dalam dokumen sejarah [3].

Pengenalan karakter selain untuk mengenali dokumen sejarah, pengenalan karakter biasanya digunakan untuk mengenali simbol pada dokumen tertentu seperti misalnya simbol pada teks Kannada. Pada pengenalan simbol teks kannada digunakan *neural classifier*. *Neural classifier* terbukti efektif dalam proses klasifikasi karakter berdasarkan fitur sesaat [4].

Sebuah algoritma pengenalan karakter diusulkan untuk mengenalai karakter Bangla yang memiliki beragam bentuk (*font*) menggunakan *Digital Curvelet Transform*. Curvelet, meskipun banyak digunakan dalam berbagai bidang pengolahan citra, belum pernah digunakan untuk ekstraksi fitur dalam pengenalan karakter. Nilai koefision *curvelet* dari suatu gambar asli serta versi *morfologis* diubah digunakan untuk melatih secara terpisah dengan menggunakan beberapa klasifier *K-Neares-Neighbor*. Nilai output dari seluruh klasifier diproses dengan menggunakan algoritma pemilihan mayoritas (voting) secara sedehana untuk diperoleh keputusan akhir [5].

ISSN 2252-3081

Metode similaritas untuk melakukan klasifikasi teks diusulkan, dengan melakukan pengukuran similaritas semantic dari dokumen teks. Tingkat similaritas dari dokumen sangat dipengaruhi oleh kata (*term*) dalam dokumen tersebut. Kata (*term*) dalam dokumen sangat bervariasi. Terdapat dua macam perhitungan similaritas semantik dari kata, yang pertama dengan menggunakan *database* kata, yang kedua dengan menggunakan pengetahuan seperti misalnya *semantic network* [6].

Cosine similarity metric learning diusulkan untuk proses pengenalan wajah. *Cosine similarity* digunakan untuk menghitung kesamaan wajah, karena pengukuran kesamaan wajah yang tepat sangat mempengaruhi proses pengenalan wajah. *Cosine similarity* digunakan untuk membawa pada sebuah algoritma pembelajaran yang efektif yang dapat meningkatkan kemampuan sendiri pada kondisi matriks yang berbeda [7].

Metode pengenalan karakter yang beragam telah diusulkan, diantaranya adalah dengan menggunakan basis data yang telah dilakukan proses optimasi. Penggunaan basis data untuk proses training tentu membutuhkan waktu yang lebih, karena proses training dilakukan secara terpisah dengan proses evaluasi. Proses training merupakan proses yang tidak terpisah dan harus dilakukan sebelum sistem melakukan proses pengenalan. *Neural Classifier* yang digunakan untuk proses pengenalan karakter juga membutuhkan proses training. Metode pengenalan karakter yang berbasis citra juga telah diusulkan dengan menggunakan *Digital Curvelet Transform* yang belum pernah digunakan pada pengolahan teks sebelumnya.

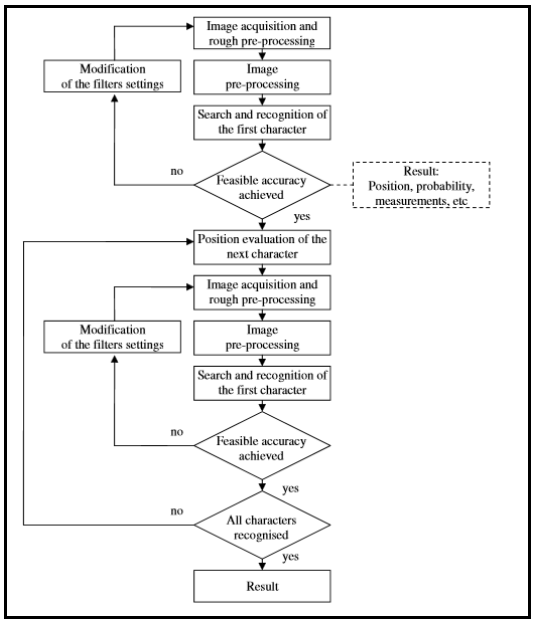
Pada makalah ini diusulkan sebuah metode pengenalan karakter berbasis citra dengan memanfaatkan algoritma similaritas menggunakan *cosine similarity*. *Cosine similarity* digunakan karena tidak membutuhkan proses learning sebelumnya. Diharapkan metode pengenalan karakter yang diusulkan, dapat memiliki nilai akurasi, presisi, dan recall yang tinggi, mendekati 100%.

2. STUDI LITERATUR

Pengenalan karakter secara harfiah diartikan sebagai sebuah proses dalam komputerisasi untuk mengenali huruf, angka, atau simbil dan mengubahnya dalam bentuk digital yang dapat digunakan untuk proses komputerisasi selanjutnya [8]. Proses pengenalan karakter memiliki masukan berupa gambar yang didalamnya mengandung karakter yang akan dikenali. Gambar dari dokumen tersebut merupakan hasil dari proses digitalisasi yaitu proses perubahan data spasial menjadi data digital. Pada proses tersebut dapat berupa proses *scanning* dengan menggunakan mesin *scanner* atau menggunakan alat optik digital yang lain seperti misalnya *digital camera*, *webcam*, atau *cctv*.

2.1. Pengenalan Karakter

Pengenalan Karakter memiliki beberapa tahapan proses. Proses awal dari pengenalan karakter merupakan proses pengolahan citra. Proses pengolahan citra dilakukan karena data masukan pada pengenalan karakter merupakan data citra digital. Seperti terlihat pada Gambar 1. Langkah pertama yang dilakukan adalah tahap pengolahan citra digital yaitu *image acquisition*, untuk mendapatkan kualitas gambar yang terbaik. *Image acquisition* dilakukan untuk menghilangkan noise yang dapat mengganggu proses ekstraksi informasi dari citra.



Gambar 1. Algoritma Umum Pengenalan Karakter [2]

Seperti yang telah dijelaskan sebelumnya, bahwa data masukan dari proses pengenalan karakter adalah citra, maka masalah yang sering terjadi adalah masalah noise dan juga distorsi jika proses digitalisasi menggunakan kamera digital. Pada Gambar 1 terlihat bahwa metode yang digunakan adalah dengan melakukan perubahan filter seperti misalnya *binarization threshold*, dan *iteration number*. Pada setiap proses iterasi menghasilkan karakter yang dibandingkan dengan contoh karakter. Metode ini diaplikasikan dengan tujuan dapat mengenali karakter yang memiliki nilai

ISSN 2252-3081

threshold yang berbeda dapat diproses dan dikenali karakternya secara tepat.

2.2. Similaritas

Similarity/similaritas jika diartikan secara harfiah merupakan kesamaan seperti misalnya didalam sebuah kalimat:"kesamaan ciri dari penyakit membuat seorang dokter sulit untuk melakukan diagnosa". Similaritas sering digunakan dalam proses pengelompokan. Similaritas digunakan untuk melakukan proses klasifikasi data yang memiliki ciri yang serupa. Data yang akan dikelompokkan dibandingkan dengan kelompok data yang sudah terdapat dalam *database* (*sample*).

Similaritas merupakan salah satu cara untuk menghitung jarak kesamaan dari dua hal yang dibandingkan, sehingga perhitungan similaritas sering dikatakan perhitungan jarak kesamaan. *Euclidean distance*, *Manhattan distance*, *hamming distance*, dan *cosine distance* merupakan beberapa contoh metode yang dapat digunakan untuk menghitung jarak kesamaan antara dua hal yang akan dibandingkan. Setiap metode memiliki masukan berupa ciri yang dapat dibandingkan, yang kemudian dapat dihitung. Nilai yang dihasilkan dari proses perhitungan similaritas menghasilkan nilai antara 0 dan 1.

2.2.1. *Euclidean Distance*

Dalam matematika, jarak *Euclidean* atau *Euclidean* metrik adalah jarak antara dua titik yang satu dengan yang lain yang dapat diukur dengna menggunakan penggaris, pada perhitungan jarak *Euclidean* digunakan formula *Pythagoras*. Dengan menggunakan formula ini sebagai jarak, ruang Euclidean menjadi ruang metrik [9]. Pada perhitungan *Euclidean distance* nilai yang didapatkan adalah nilai positif. Similaritas yang dihitung dengan menggunakan *Euclidean distance* diperoleh dengan mendapatkan nilai terendah.

Dua hal yang dibandingkan dan dihitung dengan menggunakan *Euclidean distance* dapat dikatakan mirip jika nilai yang didapatkan adalah nilai paling rendah bahkan mendekati 0. Euclidean distance dapat dihitung dengan menggunakan formula berikut ini ,

$$d(p,q) = \sqrt{\sum_{i=1}^{n} (P_i - q_i)^2}$$

Dimana:

- *p* dan *q* adalah dua buah titik yang akan dihitung jaraknya,

p_i dan q_i adalah nilai dari setiap dimensi *i* pada *p* dan *q*.

2.2.2. *Manhattan Distance*

Taksi geometri, dianggap oleh Hermann Minkowski di abad ke-19, merupakan bentuk geometri di mana fungsi jarak biasa atau metrik geometri Euclidean digantikan oleh metrik baru di mana jarak antara dua titik adalah jumlah dari perbedaan mutlak mereka Cartesian koordinat. The taksi metrik juga dikenal sebagai jarak bujursangkar, jarak L1 atau \ ell_1 norma (lihat ruang Lp), kota blok jarak, jarak Manhattan, Manhattan atau panjang, dengan variasi yang sesuai dalam nama geometri. [1] Yang terakhir nama menyinggung tata letak grid jalan paling di pulau Manhattan, yang menyebabkan jalur terpendek mobil bisa memakan waktu antara dua persimpangan di borough memiliki panjang sama dengan jarak persimpangan 'dalam geometri taksi.

2.2.3. *Hamming Distance*

Dalam teori Informasi, *Hamming Distance* antara dua *string* (kata) dengan panjang yang sama adalah jumlah posisi dimana simbol-simbol yang bersesuaian memiliki perbedaan. Dengan cara lain hal itu dapat digunakan untuk mengukur jumlah minimum substitusi yang diperlukan untuk mengubah suatu *string* ke *string* yang lain, atau jumlah minimum kesalahan yang bisa merubah satu *string* ke *string* yang lain. Seperti misalnya jika terdapat *string* seperti berikut ini [11],

- *Hamming distance* dari kata"toned" dan "roses" adalah 3.
- *Hamming distance* dari 1011101 dan 1001001 adalah 2

Hamming distance dinamai berdasarkan nama penemunya yaitu Richard Hamming, yang diperkenalkan dalam makalah *Hamming codes Error detecting and error correcting codes* pada tahun 1950 [12]. *Hamming Distance* digunakan dalam bidang telekomunikasi untuk menghitung jumlah bit yang terbalik dalam kata binner dalam jumlah panjang yang tetap sebagai perkiraan kesalahan. *Hamming Distance* digunakan dalam beberapa disiplin ilmu termasuk teori informasi, teori *coding*, dan kriptografi. Namun untuk membandigkan *string* dengan panjang yang berbeda dimana tidak hanya substitusi, namun juga penghapusan dan sisipan metric yang lebih canggih seperti *Levenstein Distance* lebih tepat digunakan [11]. Namun *Hamming Distance* juga dapat digunakan untuk menghitung jarak genetik [13].

ISSN 2252-3081

2.2.4. *Cosine Similarity*

Cosine Similarity adalah ukurang kesamaan antara dua buah vektor dalam sebuah ruang dimensi yang didapatkan dari nilai cosinus sudut dari perkalian dua buah vektor yang dibandingkan. Karena cosinus dari 0^0 adalah 1 dan kurang dari 1 untuk nilai sudut yang lain, maka nilai similaritas dari dua vektor dikatakan mirip ketika nilai dari *cosine similarity* adalah 1.

Cosine similarity terutama digunakan dalam ruang positif, di mana hasilnya dibatasi antara nilai 0 dan 1. Perhatikan bahwa batas ini berlaku untuk sejumlah dimensi, dan *Cosine similarity* ini paling sering digunakan dalam ruang positif dimensi tinggi. Misalnya, dalam *Information Retrieval*, masing-masing kata/istilah (*term*) diasumsikan sebaga dimensi yang berbeda dan dokumen ditandai dengan vektor dimana nilai masing-masing dimensi sesuai dengan berapa kali istilah muncul dalam dokumen [14].

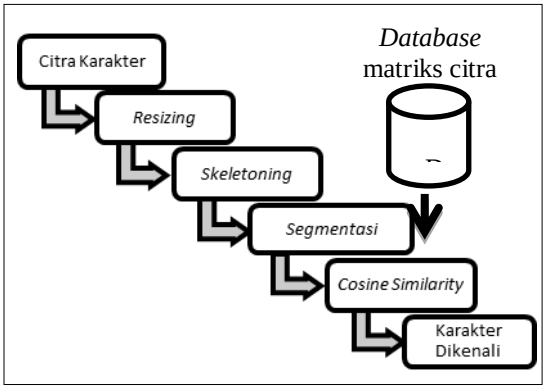
Cosine similarity dapat dihitung dengan menggunakan persamaan berikut,

$$Cosine\ Similarity = \frac{A.B}{\|A\|.\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}}$$

dimana *A* merupakan bobot setiap ciri pada vektor A, dan *B* merupakan bobot setiap ciri pada vektor B. Jika dikaitkan dengan *information retrieval* maka A adalah bobot setiap istilah pada dokumen A, dan B merupakan bobot setiap istilah pada dokumen B. Pada penelitian ini digunakan *cosine similarity* karena citra merupakan salah satu data yang memiliki dimensi tinggi. Pada citra dapat dikatakan bahwa setiap *pixel* merupakan dimensi yang berbeda dan nilai warna pada setiap *pixel* tersebut merupakan nilai dari setiap dimensi.

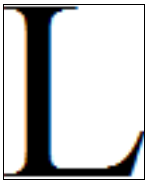
3. DESAIN METODE

Pengenalan karakter yang diusulkan merupakan proses pengenalan karakter tanpa proses *learning* terlebih dahulu. Langkah-langkah proses pengenalan karakter yang diusulkan seperti terlihat pada Gambar 2. Proses pengenalan karakter diawali dengan proses *resizing*, yaitu mengubah ukuran citra menjadi satu ukuran yang sama dengan *template* yang ada pada *database*. Proses selanjutnya adalah proses *skeletoning*, dilanjutkan dengan proses segmentasi matriks citra. Proses berikutnya adalah proses *Cosine Similarity* dengan membandingkan nilai matriks citra masukan dengan matriks citra *template* karakter yang terdapat dalam *database*.



Gambar 2. Langkah-Langkah Pengenalan Karakter

Cosine Similiarity digunakan untuk membandingkan data masukan dengan data contoh (*template*) yang telah tersimpan dalam *database*. Proses pengenalan karakter pada penelitian ini memiliki masukan berupa citra yang telah karakter yang telah tersegmentasi seperti terlihat pada Gambar 3, dimana frame dari citra merupakan frame karakter yang telah tersegmentasi. Proses segmentasi karakter dilakukan dengan mendeteksi titik kordinat terkecil dari *pixel* karakter dan titik kordinat terbesar dari *pixel* karakter.



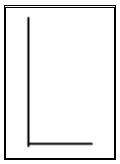
Gambar 3. Citra Masukan Untuk Uji

Dalam sistem sebelumnya telah disimpan data *template* citra dari setiap karakter yang akan diproses. Data citra yang tersimpan dalam *database* berbentuk data matrik integer, yang didalam matrik tersebut bernilai *pixel* dari setiap segmen yang terdapat pada citra *template*. Sebelum dilakukan penyimpanan data pada *database*, citra *template* terlebih dahulu diproses agar mendapatkan data template dalam bentuk matriks *integer*. Matriks *integer* yang disimpan memiliki ukuran *N x N* dengan nilai *N* sama dengan ukuran segmentasi dari citra *template* yang telah dilakukan *skeletoning*. Skeletoning dalam proses pengenalan karakter diperlukan untuk membuat suatu citra karakter memiliki ketebalan 1 (satu) *pixel* seperti terlihat pada Gambar 4.

Seperti terlihat pada Gambar 3 bahwa frame citra tetap seperti ukuran frame citra sebelum dilakukan *skeletoning*. Citra hasil *skeletoning*

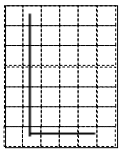
ISSN 2252-3081

selanjutnya dilakukan proses segmentasi sehingga seperti terlihat pada Gambar 5.



Gambar 4. Citra Masukan yang telah Tersegmentasi

Proses segmentasi merupakan proses membagi area citra menjadi beberapa bagian, dimana setiap bagian memiliki beberapa *pixel* dengan nilai mimal 1 (satu). Pada penelitian ini segmentasi dilakukan pada citra dengan ukuran *pixel M x N* dimana *M* dan *N* adalah konstan dengan nilai tertentu misalnya 10.



Gambar 5. Citra *Skeletoning* yang Tersegmentasi

Data akan diekstraksi dari citra yang telah tersegmentasi sehingga menjadi matriks, dengan nilai integer di setiap elemen matriks adalah jumlah *pixel* yang berwarna hitam (sesuai dengan warna *forecolour*). Segmentasi didasarkan pada ukuran citra, seperti misalnya jika ukuran citra adalah 60 x 70 *pixel*, maka pada citra tersebut dapat dilakukan segmentasi sebesar 10 x 10 sehingga membentuk matriks dengan ukuran sebesar 6 x 7 (kolom *x* baris). Setiap elemen matriks akan memiliki nilai *integer* dengan batasan antara 0 (nol) sampai dengan 100. Matriks yang terbentuk dari seperti berikut ini.

$$\begin{bmatrix} 0 & 5 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 & 0 \\ 0 & 7 & 10 & 10 & 10 & 0 \end{bmatrix}$$

Proses selanjutnya adalah proses *Cosine Similarity*, dimana setiap elemen matriks digunakan sebagai masukan yang dibandingkan dengan data matriks citra *template* pada *database*. Akhir dari proses *Cosine Similarity* adalah nilai similaritas yang didapatkan dari setiap proses perbandingan, nilai similaritas yang tertinggi dari proses perbandingan matriks masukan dengan matriks citra dalam *database* akan dipilih sebagai karakter yang dikenali.

4. SIMPULAN

Pada penelitian ini telah dibentuk sebuah metode untuk melakukan proses pengenalan karakter dengan menggunakan prinsip similaritas. Pada penelitian ini diusulkan *Cosine Similarity* sebagai proses untuk menghitung similaritas yang merupakan proses akhir pengenalan karakter. Dengan menggunakan similaritas maka proses pengenalan karakter dapat langsung dilakukan dengan menggunakan *database* matriks citra *template* dari karakter yang akan dikenali. Langkah berikutnya adalah mengimplementasikan metode yang diusulkan untuk selanjutnya dapat diuji coba.

5. Daftar Pustaka

[1] M. T. Qadri, dan M. Asif (2009), *Automatic Number Plate Recognition System for Vehicle Identification Using Optical Character Recognition*, Education Technology and Computer, 2009. ICETC '09. International Conference, Singarore.

[2] Kirill Safronov, igor Tchouchenkov, dan Heinz Wörn (2007), *Optical Character Recognition Using Optimisation Algorithms*, Proceedings of the 9th International Workshop on Computer Science and Information Technologies CSIT'2007, Ufa, Russia.

[3] Vamvakas, G.; Gatos, B.; Stamatopoulos, N.; Perantonis, S.J. (2008), *A Complete Optical Character Recognition Methodology for Historical Documents*, Document Analysis Systems, 2008. DAS '08. The Eighth IAPR International Workshop pp.525,532, 16-19.

[4] R. Sanjeev Kunte,R. D. Sudhaker Samuel (2007), *A simple and efficient optical character recognition system for basic symbols in printed Kannada text*, Sadhana Journal Volume 32, Issue 5 , pp 521-533, India.

[5] Majumdar Angshul (2007), *Bangla Basic Character Recognition Using Digital Curvelet Transform*, Journal Of Pattern Recognition Research (JPRR) 1 pp, 17-26.

ISSN 2252-3081

[6] Mihalcea Rada, Corley Courtney, dan Strapparava Carlo (2007), *Corpus-based and Knowledge-based Measures of Text Semantic Similarity*, American Association for Artificial Intelligence (www.aaai.org).

[7] V. Nguyen Hieun dan Bai Li (2010), *Cosine Similarity Metric Learning for Face Verification*, Computer Vision – ACCV 2010 Lecture Notes in Computer Science Volume 6493, 2011, pp 709-720

[8] Maccmillan Dictionary Editor (2013), *Macmillan Dictionary*, http://www.macmillandictionary.com/dictionary/british/character-recognition, diakses online 20 agustus 2013

[9] Wikipedia Editor (2013), *Euclidean Distance*, http://en.wikipedia.org/wiki/Euclidean_distance, diakses online tanggal 21 agustus 2013

[10] Eugene F. Krause (1987). *Taxicab Geometry*. Dover. ISBN 0-486-25202-7, 1987

[11] Wikipedia Editor (2013), *Hamming Distance*, http://en.wikipedia.org/wiki/Hamming_distance, diakses online tanggal 21 agustus 2013

[12] Hamming, Richard W. (1950), *Error detecting and error correcting codes*, Bell System Technical Journal **29** (2): 147–160, MR 0035935, 1950

[13] Pilcher, C. D.; Wong, J. K.; Pillai, S. K. (2008), *Inferring HIV transmission dynamics from phylogenetic sequence relationships*, PLoS Med. 5 (3): e69, doi:10.1371/journal.pmed.0050069, PMC 2267810, PMID 18351799.

[14] Singhal, Amit (2001), *Modern Information Retrieval: A Brief Overview*", Bulletin of the IEEE Computer Society Technical Committee on Data Engineering 24 (4): 35–43.

ISSN 2252-3081